Risk Management in IT Project in the Framework of Agile Development

Aneta Poniszewska-Marańda[a] and Justyna Kuna

Institute of Information Technology, Lodz University of Technology, Łódź, Poland

Keywords: Agile Approach, Agile Methodology, Scrum, Risk Management, IT Project Management.

Abstract: IT projects are increasingly complex and as a result they are subject to failure. Most of them do not meet deadlines, user requirements and run over budget. Risk management in IT projects is a crucial process, which is often understated. Agile methodologies do not give detailed guidelines for risk management. Risks are not sufficiently considered in a proactive way. As a result, there is a need to look for methods and practices possible to implement in Agile in order to improve the chances of success. Some authors suggest applying traditional practises into agile approaches. The paper introduces the risk management in agile methodologies by proposing the Scrum Risk Management process. It presents the roles, events and artefacts of the method together with the application tool as a complement to risk management process. The proposed method was also used in practice in the real IT projects.

1 INTRODUCTION

Nowadays, IT projects are increasingly complex and as a result they are subject to failure (Bannerman, 2008). Most of them do not meet deadlines, user requirements and run over budget. Risk management in IT projects is a crucial process, which is often understated. The inappropriate procedure and application of risk management activities or their absence significantly decrease the chances of success in IT projects (B.G. Tavares and de Souza, 2017). Risk management practises are the integral part of traditional project management methodologies. However, they are more and more replaced by agile approaches that are response to unpredictability of software engineering and changing business environment. There is no consensus regarding risk management within Agile. Risk management is mostly conducted in an implicit way in agile approaches. However, it is likely to ignore major risks without applying explicit risk management.

Agile methodologies do not give detailed guidelines for risk management. Risks are not sufficiently considered in a proactive way. As a result, there is a need to look for methods and practices possible to implement in Agile in order to improve the chances of success. Some authors suggest applying traditional practises into agile approaches. However, it is important to not deprive the risk management process the spirit of agility. Speaking of risk management, difference between risk and uncertainty should be emphasized. Risk is understood as uncertain event that has a positive or negative impact on the project and its likelihood can be assessed. In case of uncertainty, the probability of an event is completely unknown (Malyszek, 2020). It would be possible to prepare in advance a set of symptoms to facilitate a diagnosis and introduce measures to combat the risk, before it materializes.

This paper identifies and analyses risk management standards in traditional as well as in agile methodologies. Then, an appropriate way of risk management in Scrum, without depriving framework of characteristics and advantages, is indicated.

2 RISK MANAGEMENT IN IT PROJECTS

Risk management is a process of identification, analysis and responding to risks. Project management methodologies define a risk in different ways. According to David Hillson, risk is defined as a ”uncertainty that has an impact on project objectives” (Hillson, 2009). Risk is usually characterized by its components such as likelihood and impact that together refer to the risk exposure (Moran, 2014). Risk exposure is the basis of the standard model of risk management. This model uses the following formula: *Risk*

[a] https://orcid.org/0000-0001-7596-0813

Poniszewska-MaraÅĎda, A. and Kuna, J.
Risk Management in IT Project in the Framework of Agile Development.
DOI: 10.5220/0012127800003538
In *Proceedings of the 18th International Conference on Software Technologies (ICSOFT 2023)*, pages 540-547
ISBN: 978-989-758-665-1; ISSN: 2184-2833

factor = Likelihood x Impact.

It is possible to measure risk in costs in time or quality (D. Benţa and Mircean, 2011). In regard to project management, a risk has firstly at least one cause and then at least one effect (Benta, 2011). Archetypal risk drivers for IT projects are as follows (Moran, 2014): (1)*Requirements risks* – all risks associated with functional requirements. The problems with user acceptance can also be included in this category. The rest of non-functional requirements that are not already considered for under technical risks, can might also be incorporated here (e.g. security, usability). (2) *Technical risks* – all risks related to design, infrastructure and architecture of the developed solution. The risks are not limited to the proposed solution, but also involve dependencies (e.g. shared libraries) with the estimation of software and hardware capabilities. Generally, non-functional requirements such as scalability, performance and maintainability are included here. (3) *Schedule risks* – all risks relating to timing of activities (e.g. release planning of increments) and scheduling. It includes also financial cost consequences. (4) *Project risks* – all risks associated with the effectiveness of project management as well as levelling of resources. (5) *Supplier risks* – all risks related to external sourcing together with delivery of components (including conformity, quality and timeliness). (6) *People risks* – all risks relating to expectations of abilities and the level of skills.

The IEEE 1540 standard aims mainly at risk management in engineering. It is a response to the need for consistent terminology and framework for communicating risk in the field of engineering. It groups risk management standards within engineering to avoid redundancy. The IEEE 1540 standard ensures a minimal standard of risk management in engineering by focusing on a set of core practises (Nyfjord, 2008). Process of IEEE 1540 is shown in figure 1.

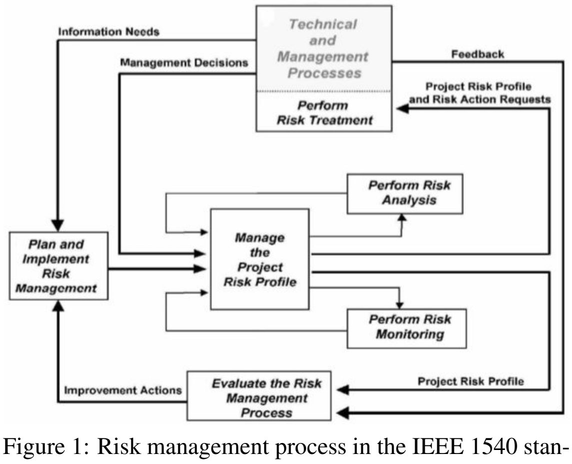

Information Needs
Technical and Management Processes
Management Decisions
Feedback
Project Risk Profile and Risk Action Requests
Perform Risk Treatment
Perform Risk Analysis
Plan and Implement Risk Management
Manage the Project Risk Profile
Perform Risk Monitoring
Improvement Actions
Evaluate the Risk Management Process
Project Risk Profile

Figure 1: Risk management process in the IEEE 1540 standard (Nyfjord, 2008).

According to Project Management Institute (PMI) following six phases of risk management are distinguished (Fig. 2) (Report, 2020): (1) *Risk management planning* – responsible for determining the approach to risk management activities as well as developing the organizational infrastructure that would support potential risk reduction, prepare alternative activities or determine cash and temporal reserves. (2) *Risk identification* – describing risk factors that may affect the project. The PMI presents many methods for collecting and using information in the field of risk recognition (brainstorming, expert judgement, root cause analysis, checklist analysis, SWOT analysis and Delphic technique) where the final result is a risk register. (3) *Qualitative risk analysis* – an assessment of risk factors, prioritizing them according to their possible impact on the project. This process helps to focus on the most important risk factors. One of the main methods of qualitative risk analysis is the matrix of the probability and impact. (4) *Quantitative risk analysis* – the probability measurement and estimation of the factors' impact on achieving the project's goals. This analysis refers not only to the existing risks but also the risk of the project presented in an all-encompassing perspective. The PMI methodology provides methods for correct quantitative risk assessment such as sensitivity analysis, analysis of expected monetary values, expert judgement, modelling and simulations. (5) *Risk response planning* – elaboration of procedures that reduce the negative impact of factors and enhance opportunities. There are several strategies resulting from an appropriate response to risks, but it is important to choose the optimal strategy to potential risk: (5.1) for threats: risk elimination, risk transfer, risk minimizing; (5.2) for opportunities: exploiting, sharing and positive effect enhancing; (5.3) for threats and opportunities: risk acceptance involving alternative plan preparation. (6) *Risk monitoring and control* – responsible for assessing the effectiveness of planned and implemented remedial actions as well as identifying new factors that appear during project realization. Following methods and tools are used: risk reassessment, technical performance measurement, variance and trends analysis, risk audits, reserve analysis and meetings.

Agile methodologies do not directly indicate the risk management process. However, some agile methodologies such as Scrum point out practical guidelines related to the environmental aspects of risk management e.g. team building. What is more, continuous improvement of processes in Agile is important for risk management (Trzeciak and Spalek, 2016). There are few characteristics of agile methodologies, which have significant impact on risk man-

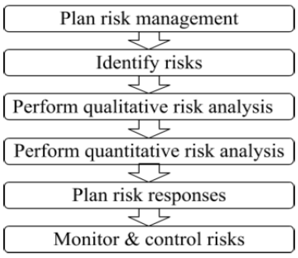


Figure 2: Risk management process in PMI methodology.

agement (Manifesto, 2001): (1) *Iterative approach* – IT projects conducted in agile methodologies, are realized in iterations. Each iteration provides functionalities that can be evaluated by a client. Then, his feedback influences next iterations and as a consequence, risk of delivering different then expected product, is decreased. (2) *Openness to changes* – due to lack of prepared realization plans for whole project like in traditional methodologies, there is no risk connected with changes in the scope of the project. Plans are created only for coming iteration, therefore there is greater confidence that they will be realized. According to Agility Manifesto, requirements can change during the project, but it does not cause an increase of risk. (3) *Self-organization of teams* – team defines what functionalities they choose and how they provide them on its own. So, risk of delays is minimized because the scope of the work is adjusted to the team.

Progressive risk reduction is a popular method often used in agile approaches, for example Scrum. This solution depends on choosing firstly these requirements that are more risky. It results in reduction of costs caused by an occurrence of risk. It is important to point out that an attempt to reduce the likelihood of risk is not included in this practise. By adding a higher priority on riskier tasks, the risk is not managed. They get addressed and completed shortly then tasks with less risk. Nevertheless, without identifying the right risks and analysing them, it is almost impossible to elaborate a plan for risk mitigation, tracking as well as controlling (Waterfall, 2020).

Despite the fact that agile models are claimed to be risk-driven, they implement only few risk management practises. Some of risk management aspects are briefly discussed on the example of Scrum framework: (1) *Risk definition* – is the same as in traditional methodologies. (2) *Risk assessment* – Scrum does not provide any guidelines at all. (3) *Roles, responsibilities* – Scrum suggest that the whole team bears the responsibility for the success of the project. However, there are no roles that could be relevant in the process of risk management. (4) *Templates* – Scrum does not provide any templates for communicating information about risk and its management. Nevertheless, high quality information about risks is one of the most crucial prerequisites for effective risk mitigation. Scrum does not contain guidance for what information should be collected and structured. (5) *Supporting tool/repository* – Scrum does not provide any recording of risk information. Traditional risk management models suggest permanent storage, whereas agile approaches prefer temporary one. Traditional risk management models recommend an experience base and risk management repository in order to assure that all significant information is being remembered and the lessons learned can be spread to enable the improvement process.

3 RELATED WORKS

Authors in (B.G. Tavares and de Souza, 2017) performed a research throughout on-site interviews conducted for 21 members of projects including Scrum Masters and Product Owners. Most of them had experience in software projects beyond 4 years and experience in Scrum more than 1 year. The aim of research was to investigate the risk management practises. The respondents had the highest agreement with the practise related to continuously performing risk management in a feedback loop. So, risk management should be updated with new project data at each iteration, and then new estimates must be generated. The second practise refers to the lessons learned or the reuse of risk knowledge. The practise in the third place refers to the need of changes in the individual behaviour and organization management. On the other hand, the practise with the lowest result refers to tendency of relegating risk management to the Scrum Master after an initiatory risk identification. The respondents claim that risk should be also managed in other Scrum events. Second not desirable practise is related to high formal planning levels. The team members believe that Scrum process must be lightweight and agile, even in projects with high risk, High-level formal planning has negative impact on agility. On the other hand, some authors note that formal documentation in Scrum projects is important. According to Read and Briggs, the lack of formal documentation makes Scrum projects harder for knowledge transfer (Read and Briggs, 2012).

Furthermore, the research of Tavares, Souza and Silva (B.G. Tavares and de Souza, 2017) shows that the Scrum is believed to have no specific activities for the risk management. 38.10% of the respondents claim that the Scrum process would be more effective by integrating traditional risk management practices.

Additionally, according to 23.81% of the respondents, the effectiveness of this integration depends on how it is performed.

Authors Nyfjord and Kajko-Mattson assert that risk management that aims to increase the chances of success in IT projects, is performed mainly in an implicit fashion in projects that use agile approaches (Nyfjord and Kajko-Mattsson, 2008). Their analysis of risk management in traditional as well as agile methodologies, results in conclusions that agile approach does not provide risk management taxonomy. They recommend incorporating traditional practises in order to ensure an effective risk management (Nyfjord and Kajko-Mattsson, 2008).

The authors of (Verma and Dhanda, 2016) describe the importance of different, the most popular currently agile software development methodologies and how the agile process can result in increasing the efficiency of the tasks in development proses. The authors of (Gold and Vassell, 2015) specify how risk management can be used to effectively balance an agile method (particularly Scrum), present the benefits or limitations encountered during the application of risk management in Scrum and deals with the other processes to effectively manage the risks during Scrum projects realization.

4 SCRUM RISK MANAGEMENT METHOD

According to a survey carried out by VersionOne (Report, 2020), Scrum is the most popular method used in organizations. However, the agile methodologies do not give detailed guidelines for risk management.

This section presents unified risk management process incorporated into Scrum framework. The proposed method called *Scrum Risk Management*, is lightweight in order to not deprive the project of agility. As Scrum framework is iterative and incremental process, risk management is also considered as a continuous and iterative event.

At the beginning of a project, *Risk Management Planning* is conducted. After functionalities are gathered and added to a Product Backlog by a Product Owner, *Risk Identification* is performed. As a consequence, new artefact called *Risk Register*, which contains identified risks is created. In next event – *Risk Assessment*, whole team assess risk parameters such as impact and likelihood. During Sprint Planning, all risks are prioritized that is their values are calculated. *Controlling risks* is a continuous event that is performed during whole Sprint. By controlling is meant tracking changes in risks and identifying new risks.

4.1 Roles

There is a need to extend Scrum Team by adding a new role – a *Risk Manager*. As a result, there are four roles in Scrum Risk Management.

Scrum Master, who is responsible for ensuring that the entire team understands agile approach, practises and principles of the Scrum Risk Management. *Development Team*, which is self-organizing and cross-functional. They are responsible for creating and delivering a Product Increment at the end of each Sprint. The Development Team chooses the number of tasks from the Product Backlog to realize during the Sprint. Each Development Team member should monitor risks connected with tasks he is responsible for. In case of observing any symptoms of risk materializing, developer should inform the rest of team about it during Daily Stand-up. It is important to emphasize that due to the principles of Scrum, whole Development Team is responsible for risk management that is identification, analysis, reactions planning and acting. *Product Owner* is responsible for managing the Product Backlog. Product Backlog items should be clearly expressed and ordered. When it comes to risk management, he should identify and monitor business risks.

The duties of *Risk Manager* include tracking the correctness of risk management, ensuring that necessary documentation is updated and if needed, inducing the team to change. This role is not related to any Scrum role. According to the idea of team self-organizing, team chooses who will fill a role of Risk Manager. What is more, they decide if this role is constant or rotary – e.g. it can be rotated between Development Team members each iteration. The division of the roles for more than one person is recommended in case of too much risk to monitor.

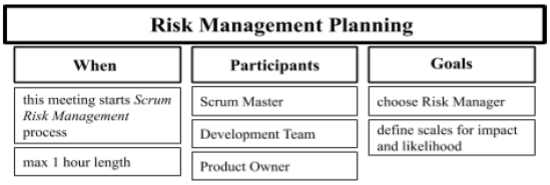
4.2 Events

The assumptions of Scrum events are not changed. Each Scrum event is time-boxed. The frame for all events is a Sprint during which a Product Increment is created. Except for Sprint, events may end if the goals are achieved. A new Sprint begins immediately after Sprint Retrospective.

In Scrum Risk Management method, risk management process is incorporated into Scrum framework. As a result, following events are proposed: (1) Risk Management Planning, (2) Risk Identification Meeting, (3) Risk Assessment Meeting, (4) Sprint Planning, which contains Risk Prioritization, (5) Daily Stand-up, (6) Sprint Review, (7) Sprint Retrospective. It is important to point out that risk controlling is performed in Sprint Planning, Daily Stand-up and Sprint Review.

4.2.1 Risk Management Planning

The maximum length for *Risk Management Planning* is 1 hour. The first step is to choose a Risk Manager. He can be changed later, however it is recommended to have a person responsible for risk management in the beginning of the project. Moreover, there are defined scales for risk parameters: impact and likelihood. The summary of Risk Management Planning is shown in figure 3.



Risk Management Planning

When	Participants	Goals
this meeting starts *Scrum Risk Management* process	Scrum Master	choose Risk Manager
max 1 hour length	Development Team	define scales for impact and likelihood
	Product Owner	

Figure 3: Overview of Risk Management Planning.

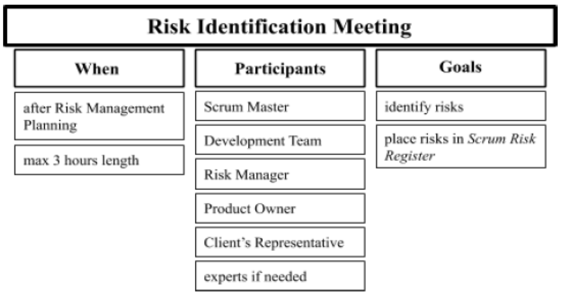
It is important to point out that technical risks should be monitored by the Development Team, while business risks should be monitored by the Product Owner.

4.2.2 Risk Identification Meeting

After finding initial product requirements, risk search should be conducted during *Risk Identification Meeting* (Fig. 4). The suggested method of risk identification is brainstorming with the participation of Development Team, Scrum Master, Risk Manager, Product Owner and Client's Representative. If specialized knowledge of any risk areas is needed, corresponding experts are invited. The participants sit together and identify the risks based on their experience with the similar projects or thinking any technical obstacles out. The good practise is to try to identify risks from the perspective of various risk categories. Scrum Risk Management proposes four categories: technical, requirements, business and process. The result of brainstorming is a list of risks, that should be placed into a new artefact called *Scrum Risk Register*. All newly identified risks have status 'open'.

4.2.3 Risk Assessment Meeting

When, all identified risks have been documented, *Risk Assessment Meeting* should be conducted (Fig. 5). Two parameters of risk: impact and likelihood must be defined by whole team. To define parameters, it is worth considering *Planning Poker*, an estimation method, widely used in agile approaches (Scrum, 2020).



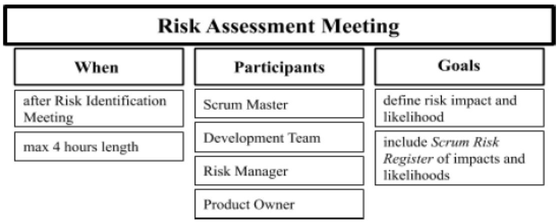
Risk Identification Meeting

When	Participants	Goals
after Risk Management Planning	Scrum Master	identify risks
max 3 hours length	Development Team	place risks in *Scrum Risk Register*
	Risk Manager	
	Product Owner	
	Client's Representative	
	experts if needed	

Figure 4: Overview of Risk Identification Meeting.

The whole Development Team participate in estimation due to the fact that each team member has different knowledge and experience that is very valuable. Product Owner, Scrum Master and Risk Manager also take part. Each Development Team member receives a set of cards with different points. They are used to determine the size of estimated impact and likelihood. The process of Planning Poker is as follows:

1. Risk Manager reads the description of risk. In order to not prolong the meeting, the good idea is to set the timeout.
2. On cue, everyone simultaneously shows chosen card with points describing firstly impact, then likelihood.
3. The difference between two consecutive values in the sequence, is a statistical error.
4. If the estimates differ significantly, team members who proposed the lowest and the highest numbers, tell why they chose such an estimate. The aim is to discover the assumptions that result in differences. Then, the vote is repeated. Usually, an agreement should be reached after two or three votes. However, if the discrepancy continues, better understanding of a given risk by team is required.



Risk Assessment Meeting

When	Participants	Goals
after Risk Identification Meeting	Scrum Master	define risk impact and likelihood
max 4 hours length	Development Team	include *Scrum Risk Register* of impacts and likelihoods
	Risk Manager	
	Product Owner	

Figure 5: Overview of Risk Assessment Meeting.

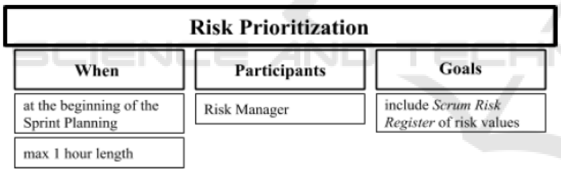
Planning Poker uses a comparison technique. However, in order to compare, a reference is required. On the beginning, there are no elements that a risk could be compared with. So, one possible solution is to take not likely risk and assign small number of points. Then, the rest risks are estimated relative to those already estimated. The recommended practice is the table of reference risks (Table 1). It is a list of few example risks each size, that is used as a reference during the discussion.

Table 1: Reference table of risk.

No	Risk description	Likelihood [in %]	Impact [1-5]	Risk value
1	Unauthorised access to confidential data	30%	1	0.3
2	Technology, tools becoming obsolete	10%	4	0.4
3	Overloading database	70%	2	1.4
4	Vendors or third-party not delivering components, tools, parts etc.	50%	3	1.5
5	Quality risks concerning writing unit, functional & automated tests for such platform	90%	5	4.5

4.2.4 Sprint Planning: Risk Prioritization

The fourth step is to prioritize risks (Fig. 6). One of the possible prioritization is the multiplication likelihood by impact. It is important to point out that for each risk parameter, there is a scale of values determined by a team. As the probability and impact increase, the value or risk also increases. *Risk Prioritization* is made at the beginning of the Sprint Planning.



Risk Prioritization

When

at the beginning of the Sprint Planning

max 1 hour length

Participants

Risk Manager

Goals

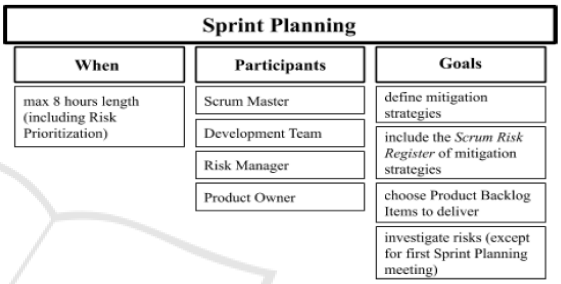
include *Scrum Risk Register* of risk values

Figure 6: Overview of Risk Prioritization during Spring Planning.

The main goal of the Sprint Planning is to determine the work to be performed during the Sprint (Fig. 7). The duration of this event should last not longer than 8 hours for a month Sprint. The whole Scrum team should take part in this meeting. Developers choose items from the Product Backlog to deliver at the end of the Sprint. Despite that, selected mitigation strategies are added to the artefact called Sprint Backlog. Splitting mitigation strategy into several tasks is possible in case of great amount of time needed to mitigate a given risk. Next reason is the situation when some tasks of a mitigation strategy should be performed in one iteration, while the rest of them in following iterations.

What is more, before choosing requirements to realize, an analysis should be performed. If a risk is likely to affect requirement realization adversely, corresponding information should be added to the Sprint Backlog. An analogous analysis is done for tasks arising from each requirement. The total value of risk connected with given requirement can be interpreted as a risk of this requirements, so it can be used for progressive risk reduction. Developer who bears responsibility for realization of given requirement or task, is also responsible for monitoring risk connected with this requirement/task.

Except for first Sprint Planning, the team is obliged to investigate all risks from *Scrum Risk Register* by answering following questions: (1) Is risk still up to date? (2) Is there any modification in risk register needed (e.g. risk re-estimation or changing status of risk)? (3) Have any new risks been identified?



Sprint Planning

When

max 8 hours length (including Risk Prioritization)

Participants

Scrum Master

Development Team

Risk Manager

Product Owner

Goals

define mitigation strategies

include the *Scrum Risk Register* of mitigation strategies

choose Product Backlog Items to deliver

investigate risks (except for first Sprint Planning meeting)

Figure 7: Overview of Spring Planning.

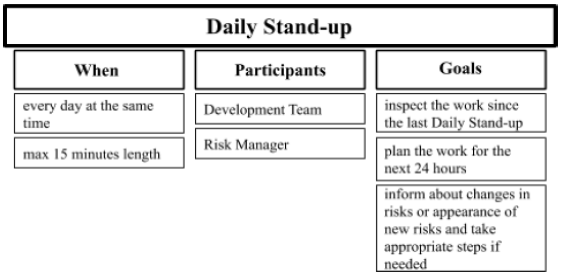
4.2.5 Daily Stand-ups

The main principles of *Daily Stand-ups* are not changed. It is at most 15 minute meeting that is held every day of the Sprint at the same time. The aim of this event is to inspect the work since the last Daily Scrum and to plan the work for the next 24 hours.

Team is not obligated to investigate all risks from *Scrum Risk Register* as it is performed in the Sprint Planning. However, if team member observed any change in risk or appearance new risks, he should inform others about it during Daily Stand-up. If one of above situations took place, appropriate steps should be taken same day. Team should evaluate new risk or take actions in response to risk using guidelines from *Scrum Risk Register*. In case of starting mitigation strategy, status of risk should be changed to 'being mitigated'. An overview of Daily Stand-up is presented in figure 8.

4.2.6 Sprint Review

The length of a *Sprint Review* is maximum 4 hours for a one-month Sprint. All Team members with stakeholders should take part in the event. The aim of this event is to inspect the Increment and if needed, adapt the Product Backlog in order to optimize the value

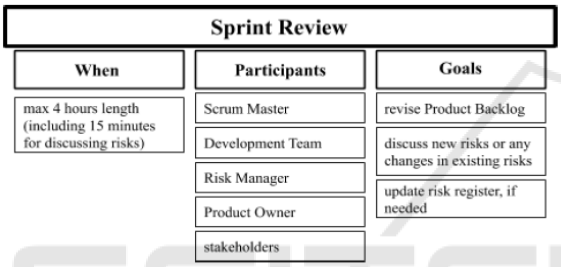


Daily Stand-up

When	Participants	Goals
every day at the same time	Development Team	inspect the work since the last Daily Stand-up
max 15 minutes length	Risk Manager	plan the work for the next 24 hours
		inform about changes in risks or appearance of new risks and take appropriate steps if needed

Figure 8: Overview of Daily Stand-up.

(Fig. 9). During the Sprint Review, the whole team should spend about 15 minutes discussing new risks or any changes in existing risks. *Scrum Risk Register* should be updated, if needed. If some risks do not have prepared mitigation strategies, it is needed to do it in the subsequent iteration.



Sprint Review

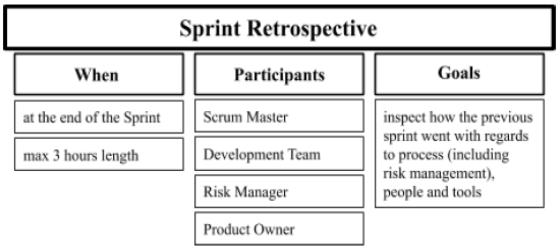
When	Participants	Goals
max 4 hours length (including 15 minutes for discussing risks)	Scrum Master	revise Product Backlog
	Development Team	discuss new risks or any changes in existing risks
	Risk Manager	update risk register, if needed
	Product Owner	
	stakeholders	

Figure 9: Overview of Sprint Review.

4.2.7 Sprint Retrospective

Length of a Sprint Retrospective is maximum 3 hours for a one-month Sprint. The aim of this meeting is inspection how the previous Sprint went with regards to process, relationships and tools (Fig. 10).

During Retrospective, an evaluation of risk management is performed. Each team member has a chance to speak about his objections regarding risk management, as well as to propose some improvements. Then, chosen changes confirmed by team are implemented in the next iteration.



Sprint Retrospective

When	Participants	Goals
at the end of the Sprint	Scrum Master	inspect how the previous sprint went with regards to process (including risk management), people and tools
max 3 hours length	Development Team	
	Risk Manager	
	Product Owner	

Figure 10: Overview of Sprint Retrospective.

4.3 Artifacts

The artifacts of Scrum Risk Management method are as follows:

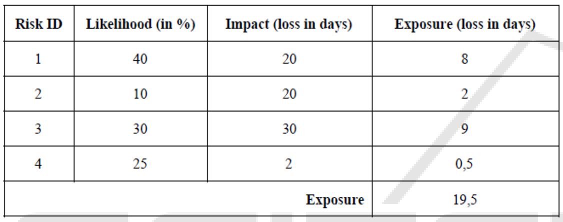
- *Product Backlog* – list of everything needed for product creation (requirements, enhancements and fixes).
- *Sprint Backlog* – a part of Product Backlog that is selected during Sprint Planning. It also contains mitigation strategies.
- *Sprint Goal* – an objective for the Sprint, can be fulfilled through implementation of the Product Backlog.
- *Increment* – sum of all completed Sprint Backlog Items during given Sprint plus values of Increments from the previous Sprints. In the context of risk management, the Increment is crucial artifact, because reduces the probability of project failure by providing quick feedback from customer. It also helps in identifying new risks and implementing their treatment. Furthermore, the Increment represents milestones of project, which should be controlled. It evaluates the feasibility of cost, quality and deadline. During creating the Increment, it is important to focus on deliverables that are clearly defined and to make frequent deliveries.
- *Definition of ”Done”* determines what it means that work is completed on the product Increment.

Scrum Risk Management method introduces two new artifacts *Scrum Risk Register* and *Risk Repository*. The aim of Scrum Risk Register is storing information about risks. It should contain following attributes: (a) *Risk ID* – an identification number of risk, (b) *Risk description* – a short overview of risk, (c) *Risk category* – a category of risk: requirements, technical, business or process, (d) *Sprint number* [optional] – indicates a sprint/sprints when risk will probably occur, (e) *User story* [optional] – a user story connected with given risk, (f) *Date identified* – date of risk identification, (g) *Likelihood* – probability of risk occurrence, (h) *Impact* – a consequence of an event, (i) *Priority* – any chosen, independent value or the product of likelihood and impact (in other words risk value), (j) *Owner* – person responsible for given risk, (k) *Risk symptoms* [optional] – symptoms that can help in early risk materialization, (l) *Mitigation strategy* – a response to given risk, (m) *Status* – a current status of risk: open, being mitigated, completed.

It is also a good idea to start making the Risk Repository. It is a list of risks that occurred in the past projects. Risk repository helps to find mitigation mechanisms by relating the risks with the previous ones.

4.4 Tools

It is also recommended to use burndown chart to analyse and assess the risk. It indicates a drop in exposition to the risk across iterations. Following input data is required: (a) item the list of risks, (b) likelihood, (c) impact (loss in days was assumed), (d) risk value being the product of likelihood and impact (assuming the number of days of exposure to the risk). Example of input data to the burndown chart is illustrated in figure 11. This simple risk census shows how a given risk is likely to happen, the size of loss if the risk occurs and the the prospective risk exposure values. It is needed to remember, that above values can change over the course of Sprints.



Risk ID	Likelihood (in %)	Impact (loss in days)	Exposure (loss in days)
1	40	20	8
2	10	20	2
3	30	30	9
4	25	2	0,5
		Exposure	19,5

Figure 11: Exemplary input data to burndown chart.

Scrum Risk Management is a unified risk management process incorporated into Scrum framework. Some activities of traditional risk management process were included. However, new method proposal is iterative, lightweight and does not deprive a project of agility. Scrum Risk Management proposes new events, artifacts and role. Particularly important is the Scrum Risk Register, which stores the information about risks.

5 CONCLUSIONS

This paper presented the new proposal of Scrum Risk Management approach. It has been proved that not only the introduction of risk management to agile approach is necessary, but also possible. Proposed method is fully integrated with Scrum framework in order to keep agile features. Explicit risk management activities have been added to tasks that already exist in agile process. Scrum Risk Management method was used for developing IT Internship Search portal, allowing adding, filtering and viewing internship offers. The progress of the project throughout its iterations was described. We demonstrated that Scrum Risk Management method contributes to the success of the project, since risks were identified and mitigated in early phases. After third Sprint, all risks from the Scrum Risk Register were mitigated.

REFERENCES

Bannerman, P. (2008). Risk and risk management in software projects: a reassessment.

Benta, D. (2011). *Software development and risk management in complex projects, Phd Thesis Summary*. Babes-Bolyai University of Cluj-Napoca.

B.G. Tavares, E. S. d. S. and de Souza, A. D. (2017). Risk management analysis in scrum software projects.

D. Benţa, M. P. and Mircean, C. (2011). On best practices for risk management in complex projects.

Gold, B. and Vassell, C. (2015). Using risk management to balance agile methods: A study of the scrum process. In *Proc. of 2nd Intern. Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 49–54*. IEEE.

Hillson, D. (2009). *Managing risk in projects*. Gower.

Malyszek, E. (2020). *Project Management in micro and small enterprises*. Online, http://www.lbs.pl/projekt/dobrepraktyki2011/files/artykuly/Malyszek.pdf.

Manifesto, A. (2001). *Agile Manifesto*. Online, https://agilemanifesto.org/.

Moran, A. (2014). *Agile Risk Management*. Springer.

Nyfjord, J. (2008). *Towards integrating agile development and risk management*. Department of Computer and Systems Sciences.

Nyfjord, J. and Kajko-Mattsson, M. (2008). Outlining a model integrating risk management and agile software development. In *Proc. of 34th EUROMICRO Confer. on Software Engineering and Advance Applications, IEEE, Italy, pp. 476–483*.

Read, A. and Briggs, R. (2012). The many lives of an agile story: design processes, design products, and understandings in a large-scale agile development project. In *45th Hawaii Intern. Confer. on System Science, HICSS, pp. 5319–5328*.

Report, A. (2020). *13th Annual State of Agile Report*. Online, https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508.

Scrum (2020). *Planning Poker cards*. Online, https://www.scrumdesk.com/agile-estimation-principles/planning-poker-cards/.

Trzeciak, M. and Spalek, S. (2016). Risk management within traditional and agile methodologies in project management.

Verma, B. and Dhanda, M. (2016). A review on risk management in software projects.

Waterfall (2020). *Waterfall project management*. Online.